

Informe de investigación.

Alumnos:

- **José Manuel Cortes Cerón**
- **Carmen Anahí Cornejo López**
- **Gustavo Barrera Martínez**

**Instituto Tecnológico Superior del
Occidente del Estado de Hidalgo**

Informe de investigación.

Materia: Proyecto Integrador

Docente: **Pedro Jhoan Salazar Pérez**

Semestre: 2 Grupo: "A"

Ingeniería en Tecnologías de la Información y
Comunicaciones

Fecha de entrega: 6 de marzo, 2024

Índice de contenido:

Informe general	3
Condiciones óptimas para su cultivo	4
Variables para el crecimiento de lechugas	4
Proceso de cultivo	5
Tabla de crecimiento de la lechuga, primeros 14 días. (Germinación)	7
Tabla de crecimiento de la lechuga (Etapas de crecimiento hasta punto máximo del crecimiento)	7
Mediciones	8
Medición de Humedad y Temperatura en el Cultivo de Lechugas Romanas	9
El electromagnetismo en la agricultura	10
Acerca de las antenas de cobre	11
Hipótesis	11
Fotografías del proyecto:	12
Informe de Desarrollo: Sistema de Control de Siembras de Lechugas	15
Imágenes del programa	18
Cálculo de volumen de la lechuga	23
Conclusiones	25

Informe general

Descripción botánica: La lechuga romana tiene hojas alargadas y crujientes, con una cabeza densa y alargada en forma de roseta. A menudo se la conoce como "lechuga cos" debido a su origen en la región del Mediterráneo y su uso tradicional en ensaladas tipo Caesar.

Cultivo: Es una planta de clima fresco que prefiere temperaturas moderadas, generalmente entre 15°C y 25°C. Se puede cultivar tanto en el suelo como en contenedores, y requiere suelos bien drenados y ricos en materia orgánica. La lechuga romana puede tolerar cierta sombra, pero prefiere la luz solar directa para un crecimiento óptimo.

Cosecha: Las hojas exteriores se pueden cosechar a medida que la planta crece, o se puede esperar a que la cabeza esté completamente formada para cosecharla en su totalidad. La lechuga romana es una planta de crecimiento rápido, por lo que se puede cosechar en aproximadamente 60-80 días después de la siembra, dependiendo de las condiciones de crecimiento.

Nutrición: La lechuga romana es baja en calorías y una buena fuente de fibra, vitaminas y minerales. Es especialmente rica en vitamina A y vitamina K, y también proporciona vitamina C, calcio, hierro y potasio.

Usos culinarios: La lechuga romana es popular en ensaladas, sándwiches, wraps y platos de inspiración mediterránea. Tiene un sabor suave y crujiente que complementa una variedad de ingredientes y aderezos.

Beneficios para la salud: La lechuga romana es una opción saludable y nutritiva para incluir en la dieta. Además de su bajo contenido calórico, su alto contenido de agua y fibra la convierten en un alimento saciante y beneficioso para la digestión. También se ha asociado con beneficios para la salud del corazón y la salud ocular debido a su contenido de nutrientes antioxidantes.

Condiciones óptimas para su cultivo

Temperatura (°C)	Humedad (%)	PH del Suelo	Mejor Abono Orgánico	Estación/es del Año	Hemisferio
15-25°C	40-60%	6,0-7,0	Compost, estiércol o vermicompost	Primavera y Otoño	Ambos

Variables para el crecimiento de lechugas

1. Temperatura:

- La temperatura ambiente afecta significativamente el crecimiento de la lechuga romana. La temperatura óptima para su crecimiento está generalmente entre 15°C y 25°C.

2. Humedad del suelo:

- La humedad del suelo es crucial para el crecimiento de la lechuga. Demasiada o muy poca agua puede afectar negativamente el desarrollo de la planta.

3. Longitud:

- El tamaño, o longitud, de la lechuga romana es una variable crítica que refleja su crecimiento y desarrollo.

Proceso de cultivo

El proceso de cultivo de la lechuga romana es bastante similar al de otras variedades de lechugas. Un entorno fresco, un suelo fértil y un suministro adecuado de agua son requisitos esenciales. Si se opta por el cultivo de esta lechuga en un invernadero, a continuación, se presentan algunas pautas para lograr una cosecha abundante:

1. Selección de la Variedad y Temporización:

- La elección de una variedad de lechuga que se adapte a las condiciones climáticas locales es fundamental. Dado que este tipo de lechuga requiere temperaturas más frías para su desarrollo, se recomienda su cultivo en otoño e invierno en zonas de clima templado. En regiones más cálidas, es más adecuado su cultivo en primavera o verano.

2. Siembra de Semillas:

- La siembra de las semillas debe realizarse en una bandeja que contenga sustrato específico para este propósito. Se debe regar con moderación para mantener la humedad del suelo, evitando el encharcamiento. La ubicación de la bandeja en un lugar fresco y soleado, cubriéndola con plástico para conservar la humedad, resulta crucial en esta etapa.

3. Germinación y Cuidado de Plántulas:

- Cuando las plantas comiencen a germinar, se debe retirar el plástico. El riego diario es necesario para mantener la humedad del suelo, evitando el exceso de agua.

4. Trasplante:

- El trasplante de las plántulas a macetas o jardineras se realiza después de alcanzar una edad de 2-3 semanas. Es importante elegir recipientes lo suficientemente espaciosos para permitir el desarrollo de las lechugas, pero no tan grandes como para provocar el encharcamiento del sustrato con el riego. El riego diario es esencial para mantener la humedad del suelo.

5. Cuidado Continuo:

- La eliminación regular de las malas hierbas y un riego constante para mantener el suelo húmedo, pero no encharcado, son aspectos cruciales a tener en cuenta. Las lechugas romanas estarán listas para su recolección aproximadamente entre 8 y 10 semanas después del trasplante.

El proceso de cultivo de la lechuga romana comienza con la selección de un lugar que reciba una generosa cantidad de luz solar. Es crucial preparar el suelo para que sea fértil, mezclándolo con compost para enriquecerlo.

Posteriormente, las semillas se siembran a una profundidad de aproximadamente 5 cm en hoyos preparados previamente.

La germinación de las plantas se observa generalmente en un lapso de 10 a 14 días.

Una vez que las plántulas han brotado, se recomienda trasplantarlas a macetas o jardineras adecuadas. El riego diario resulta esencial para mantener la humedad del suelo en niveles óptimos.

Después de un período de entre 50 y 60 días, las lechugas romanas estarán listas para su cosecha.

Este tiempo de crecimiento les permite alcanzar la madurez ideal para su consumo. El cuidado adecuado y la observación de estos pasos en el proceso de cultivo asegurarán una cosecha exitosa y satisfactoria.

**Tabla de crecimiento de la lechuga, primeros 14 días.
(Germinación)**

Dia	Temperatura (°C)	Humedad (%)	Longitud

**Tabla de crecimiento de la lechuga (Etapa de crecimiento
hasta punto máximo del crecimiento)**

Dia	Temperatura (°C)	Humedad (%)	Longitud

Mediciones

Durante la etapa de germinación, es especialmente importante proporcionar condiciones óptimas de humedad para que las semillas de lechuga romana puedan brotar y establecerse correctamente. Aquí hay algunas pautas específicas para regar durante esta etapa inicial:

- **Mantener el suelo constantemente húmedo:** Durante la germinación, asegúrate de que el suelo esté constantemente húmedo, pero no empapado. Esto ayudará a proporcionar el entorno adecuado para que las semillas germinen y las plántulas se desarrollen.
- **Regar con cuidado:** Utiliza un rociador o una regadera con una boquilla suave para regar con cuidado las semillas y el suelo circundante. Evita regar con demasiada fuerza, ya que esto podría desplazar las semillas o compactar el suelo.
- **Evitar el encharcamiento:** Aunque es importante mantener el suelo húmedo, evita el encharcamiento, ya que esto puede provocar la pudrición de las semillas o de las plántulas recién emergidas.
- **Cubrir el suelo:** Puedes cubrir las semillas recién sembradas con una fina capa de mantillo o vermiculita para ayudar a retener la humedad en el suelo y proteger las semillas de la desecación.
- **Monitorear la humedad del suelo:** Utiliza un medidor de humedad del suelo o simplemente verifica la humedad del suelo tocándolo con los dedos para asegurarte de que se mantenga adecuadamente húmedo.

Medición de Humedad y Temperatura en el Cultivo de Lechugas Romanas

En el contexto de nuestro proyecto de investigación sobre el cultivo de lechugas romanas, hemos implementado un sistema de monitoreo de humedad y temperatura utilizando un sensor DHT22 conectado a un Arduino Uno R3. El objetivo de esta implementación es realizar un seguimiento preciso de las condiciones ambientales en el entorno de cultivo, con el fin de entender mejor el impacto de estas variables en el crecimiento y desarrollo de las plantas.

El sensor DHT22, también conocido como AM2302, es capaz de medir tanto la humedad relativa como la temperatura ambiente con una precisión razonable. A través de la integración de este sensor con un microcontrolador Arduino Uno R3, podemos obtener lecturas en tiempo real de estas variables y registrarlas para su posterior análisis.

El código implementado en el Arduino Uno R3 utiliza la biblioteca de Adafruit para interactuar con el sensor DHT22. En el programa, se realiza una inicialización del sensor en la función `setup()`, seguida de un bucle principal en la función `loop()` donde se realizan las lecturas de humedad y temperatura cada 10 segundos. Estas lecturas se imprimen en el puerto serial junto con el índice de calor calculado para proporcionar una visión completa de las condiciones ambientales.

Durante la etapa de germinación de las lechugas romanas, es crucial mantener un ambiente óptimo de humedad y temperatura para facilitar la germinación y el establecimiento inicial de las plántulas. Con nuestro sistema de monitoreo en su lugar, podemos asegurarnos de que estas condiciones sean adecuadas y realizar ajustes según sea necesario para optimizar el crecimiento y la salud de las plantas.

El uso de tecnología de monitoreo como el sensor DHT22 y Arduino no solo nos permite obtener datos precisos y en tiempo real sobre las condiciones ambientales en nuestro entorno de cultivo, sino que también nos brinda la oportunidad de automatizar el proceso de recolección de datos, lo que ahorra tiempo y esfuerzo en comparación con los métodos de monitoreo manual tradicionales.

En resumen, la implementación de nuestro sistema de monitoreo de humedad y temperatura es una herramienta invaluable en nuestro proyecto de investigación sobre el cultivo de lechugas romanas. Proporciona una base sólida para comprender mejor las condiciones ambientales en el entorno de cultivo y su impacto en el

crecimiento de las plantas, lo que nos permite tomar decisiones informadas para mejorar la productividad y la calidad de nuestros cultivos.

El electromagnetismo en la agricultura

1. Antecedentes: La idea de utilizar corrientes eléctricas en la agricultura no es nueva. La electrofisiología vegetal ha sido objeto de estudio durante décadas, y se han investigado diferentes métodos para aplicar corrientes eléctricas en plantas con el objetivo de mejorar su crecimiento, resistencia a enfermedades y rendimiento.

2. Principios de la técnica: La siembra de vegetales utilizando antenas de cobre se basa en la hipótesis de que la aplicación de corrientes eléctricas de baja intensidad puede estimular el metabolismo de las plantas y promover su crecimiento. Se cree que estas corrientes eléctricas pueden afectar la absorción de nutrientes, la actividad enzimática y otros procesos fisiológicos en las plantas, lo que potencialmente podría aumentar su rendimiento y calidad.

3. Implementación práctica: Para implementar esta técnica, se diseñaron antenas de cobre que capturen la energía solar y la conviertan en corrientes eléctricas de baja intensidad. Estas antenas se colocarían estratégicamente en el entorno de cultivo, proporcionando una fuente continua de estimulación eléctrica a las plantas.

4. Posibles beneficios: Entre los posibles beneficios de la siembra de vegetales utilizando esta técnica se incluyen:

- Mejora del crecimiento y desarrollo de las plantas.
- Aumento del rendimiento y la calidad de los cultivos.
- Mayor resistencia a enfermedades y estrés ambiental.
- Reducción de la dependencia de fertilizantes y productos químicos.

5. Consideraciones y desafíos: Es importante tener en cuenta que esta técnica aún está en fase experimental y presenta varios desafíos y consideraciones a tener en cuenta:

- Es necesario investigar más a fondo para comprender completamente los efectos de la estimulación eléctrica en las plantas y su interacción con otros factores ambientales.
- Se requiere un diseño cuidadoso de las antenas y la intensidad de las corrientes eléctricas para evitar posibles efectos negativos en las plantas o en el suelo.
- Los costos de implementación y mantenimiento de esta técnica pueden ser significativos y deben ser evaluados en relación con los posibles beneficios.

Acerca de las antenas de cobre

1. Tamaño de las Antenas: El tamaño de las antenas puede afectar su capacidad para capturar energía solar y convertirla en corrientes eléctricas. Antenas más grandes tendrán una mayor área de superficie expuesta al sol, lo que potencialmente resultará en una mayor captura de energía. Sin embargo, el tamaño de las antenas también puede ser un factor limitante en términos de espacio disponible en el entorno de cultivo. Es importante encontrar un equilibrio entre el tamaño de las antenas y su eficiencia en la captura de energía solar.

2. Número de Vueltas del Enrollado: El número de vueltas del enrollado se refiere a la cantidad de vueltas de cable de cobre alrededor de la estructura de la antena. Este factor puede afectar la intensidad de las corrientes eléctricas generadas. Un mayor número de vueltas puede aumentar la intensidad de las corrientes, lo que potencialmente podría tener un mayor efecto en las plantas. Sin embargo, un exceso de vueltas puede aumentar la resistencia eléctrica y disminuir la eficiencia de la antena. Es importante encontrar un equilibrio óptimo en el número de vueltas para maximizar la eficacia de la antena.

Hipótesis

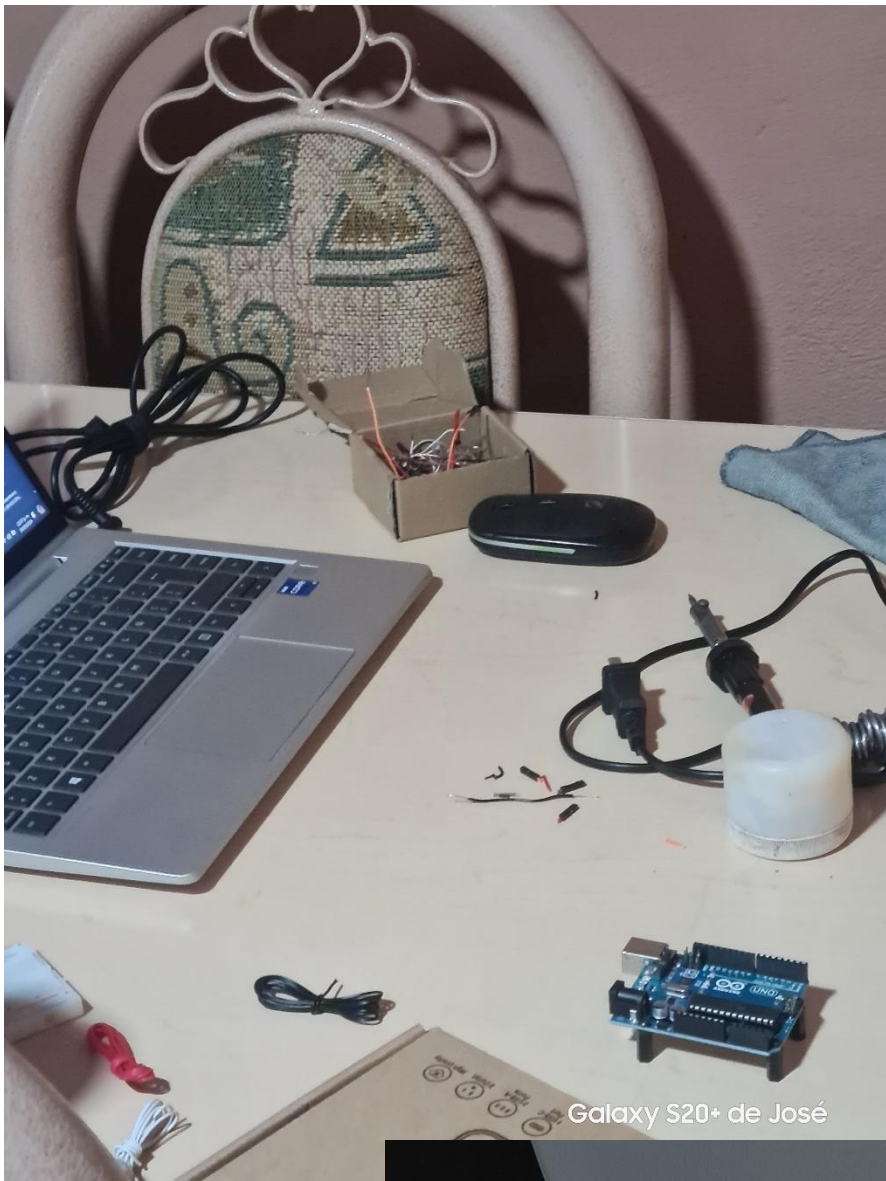
Se plantea que la aplicación de corrientes eléctricas de baja intensidad en la agricultura, mediante el uso de antenas de cobre, puede tener un impacto positivo en el crecimiento, rendimiento y resistencia de los cultivos. Esta técnica se fundamenta en la estimulación del metabolismo vegetal a través de la manipulación eléctrica, lo que potencialmente mejora la absorción de nutrientes, la actividad enzimática y otros procesos fisiológicos clave. Sin embargo, para maximizar su efectividad, es crucial encontrar un equilibrio entre el tamaño de las antenas y el número de vueltas del enrollado, optimizando así la captura de energía solar y la generación de corrientes eléctricas sin afectar negativamente a las plantas o al suelo. Aunque aún se encuentre en fase experimental, esta técnica podría ofrecer una alternativa prometedora para mejorar la productividad agrícola y reducir la dependencia de fertilizantes y productos químicos, siempre y cuando se aborden los desafíos y consideraciones asociados.

Fotografías del proyecto:

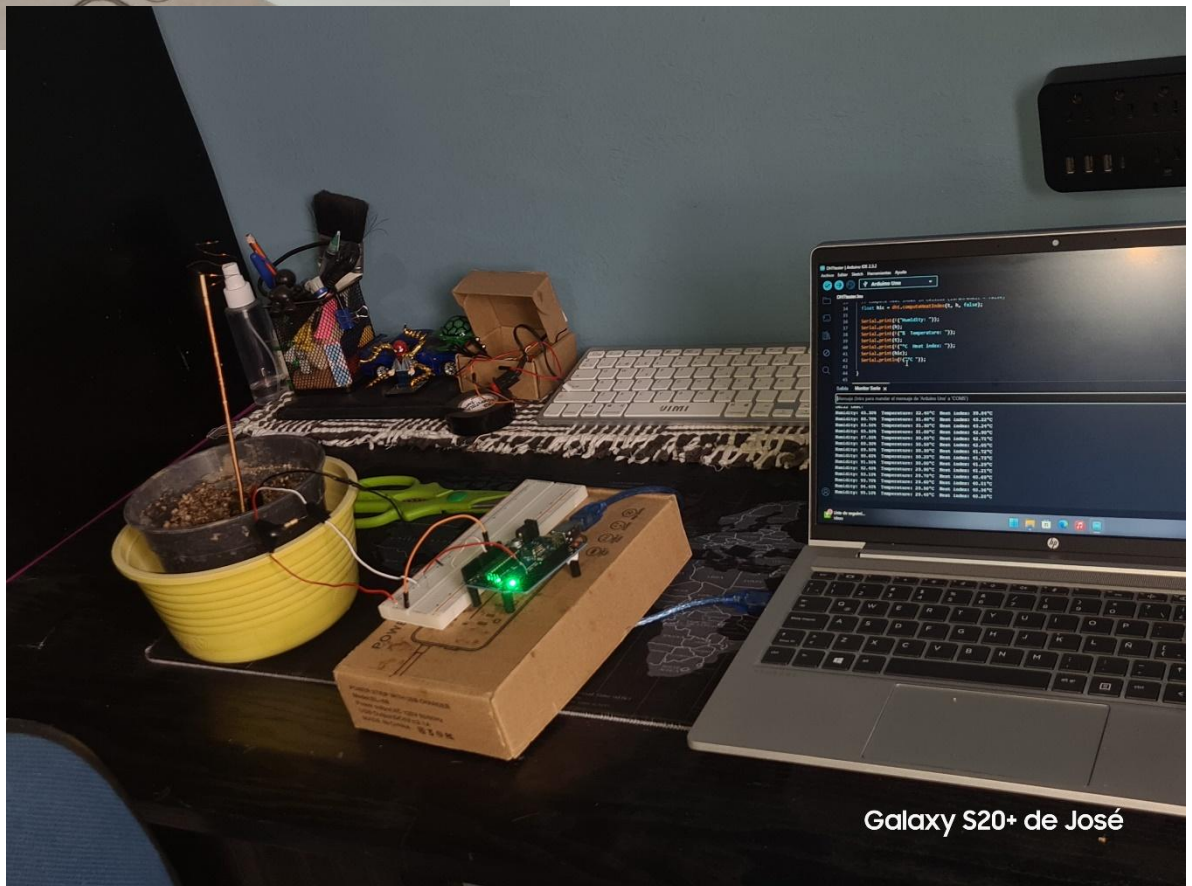




Galaxy S20+ de José



Galaxy S20+ de José



Galaxy S20+ de José

Informe de Desarrollo: Sistema de Control de Siembras de Lechugas

Resumen

Este informe detalla el desarrollo de un sistema de control de siembras de lechugas, implementado en Java utilizando el paradigma de programación orientada a objetos. El sistema permite al usuario registrar información sobre lechugas, como la fecha de plantación, la temperatura, la humedad y la longitud, y almacena estos datos en un archivo de texto para su posterior consulta.

Objetivos del Proyecto

El objetivo principal del proyecto fue desarrollar una aplicación que facilitara el registro y control de siembras de lechugas de manera eficiente y organizada. Los objetivos específicos incluyeron:

1. Crear una interfaz de usuario intuitiva utilizando la biblioteca Swing de Java.
2. Implementar clases orientadas a objetos para representar las lechugas y el registro de lechugas.
3. Permitir al usuario ingresar información sobre lechugas y registrarla en un archivo de texto.
4. Proporcionar funcionalidades adicionales, como ver el registro de lechugas, vaciar los campos del formulario y eliminar todos los registros almacenados.

Desarrollo del Programa

El desarrollo del programa se dividió en varias etapas:

1. Diseño de Clases

Se diseñaron dos clases principales:

- **Lechuga:** Representa una lechuga con atributos como ID, fecha de plantación, temperatura, humedad y longitud. Se implementaron métodos para obtener información y el ID de la lechuga.
- **RegistroLechugas:** Representa un registro de lechugas, que mantiene una lista de objetos de tipo Lechuga. Se implementaron métodos para agregar una lechuga al registro y para guardar el registro en un archivo de texto.

2. Implementación de la Interfaz de Usuario

Se desarrolló una interfaz de usuario utilizando la biblioteca Swing de Java. El formulario permite al usuario ingresar información sobre lechugas, como la fecha de plantación, la temperatura, la humedad y la longitud, y proporciona botones para registrar esta información, ver el registro de lechugas, vaciar los campos del formulario y eliminar todos los registros almacenados.

3. Integración de la Lógica de Negocios

Se integró la lógica de negocios del programa con la interfaz de usuario. Se escribieron controladores de eventos para manejar las acciones del usuario, como hacer clic en el botón de registro, ver el registro de lechugas o eliminar registros. Estos controladores interactúan con las clases Lechuga y RegistroLechugas para realizar las operaciones correspondientes, incluyendo la escritura de datos en un archivo de texto.

4. Encriptación de los datos

Para asegurar la integridad y confidencialidad de la información almacenada en el sistema, se implementó la funcionalidad de encriptación de datos utilizando el algoritmo AES (Advanced Encryption Standard). AES ha sido ampliamente reconocido como un pilar fundamental en el ámbito del cifrado simétrico y ha sido adoptado como estándar de referencia por el Instituto Nacional de Estándares y Tecnología (NIST) de los Estados Unidos desde 2001.

Este algoritmo opera sobre bloques de datos de 128 bits, ofreciendo flexibilidad en la longitud de las claves utilizadas, que pueden ser de 128, 192 o 256 bits. La flexibilidad en la longitud de las claves permite a los usuarios adaptar la seguridad de sus sistemas según las necesidades específicas de cada aplicación.

El proceso de cifrado de AES se basa en la realización de múltiples rondas de transformaciones, cuya cantidad varía dependiendo de la longitud de la clave empleada. Para claves de 128 bits, se ejecutan 10 rondas; para claves de 192 bits, se realizan 12 rondas; y para claves de 256 bits, se llevan a cabo 14 rondas. Estas rondas de transformaciones incluyen operaciones como SubBytes, ShiftRows, MixColumns y AddRoundKey, que alteran los datos de manera específica en cada etapa del proceso, proporcionando así un alto nivel de seguridad.

El proceso de descifrado sigue un procedimiento similar al de cifrado, pero en orden inverso, permitiendo recuperar el texto original a partir del texto cifrado mediante el uso de la clave de descifrado correspondiente.

La robustez y eficiencia de AES lo convierten en una opción ampliamente utilizada en diversas aplicaciones, desde el cifrado de datos en dispositivos electrónicos hasta la protección de comunicaciones en línea. Su capacidad para garantizar la confidencialidad e integridad de la información lo posiciona como un componente fundamental en la seguridad de sistemas y redes en la era digital. En el contexto de este proyecto, la implementación de AES asegura la protección de los datos almacenados, brindando tranquilidad y confianza a los usuarios del sistema de control de siembras de lechugas.

5. Pruebas y Depuración

Se realizaron pruebas exhaustivas del programa para garantizar su funcionalidad y estabilidad. Se corrigieron errores y se depuró el código según fuera necesario para garantizar un funcionamiento óptimo.

Conclusiones

El desarrollo de este sistema de control de siembras de lechugas proporcionó una experiencia práctica en la implementación de aplicaciones orientadas a objetos en Java. Se lograron los objetivos del proyecto y se creó una herramienta funcional y fácil de usar para el registro y control de siembras de lechugas. La capacidad de almacenar los datos en un archivo de texto proporciona una forma conveniente de mantener un registro de las siembras realizadas.

Futuras Mejoras

Se identificaron varias áreas para futuras mejoras:

- Implementación de validaciones de entrada más robustas para garantizar la integridad de los datos ingresados por el usuario.
- Mejora de la interfaz de usuario con características adicionales, como gráficos y estadísticas sobre las siembras de lechugas. (Cálculo de solido de revolución)

Imágenes del programa

Clase “Lechuga”:

```
1 package classes;
2
3 /**
4  *
5  * @author Josee
6  */
7
8 public class Lechuga {
9
10     private int id;
11     private String fechaPlantacion;
12     private double temperatura;
13     private double humedad;
14     private String longitud;
15
16     public Lechuga(int id, String fechaPlantacion, double temperatura, double humedad, String longitud) {
17
18         this.id = id;
19         this.fechaPlantacion = fechaPlantacion;
20         this.temperatura = temperatura;
21         this.humedad = humedad;
22         this.longitud = longitud;
23     }
24
25     // Método para obtener la información de la lechuga
26     public String getInfo() {
27
28         // Devuelven en el archivo una cadena de texto
29         return "ID: " + id + ", Fecha de Plantación: " + fechaPlantacion + ", Temperatura: " + temperatura + " °C, Humedad: " + humedad + "%, Longitud: " + longitud + " cm";
30     }
31
32     // Método para obtener el ID de la lechuga
33     public int getId() {
34         // Retornar el ID
35         return id;
36     }
37 }
38
```

Clase RegistroLechuga:

```
1 package classes;
2
3 import java.io.FileWriter;
4 import java.io.IOException;
5 import java.io.PrintWriter;
6 import java.util.ArrayList;
7
8 /**
9  *
10  * @author Josee
11  */
12
13 public class RegistroLechugas {
14     private ArrayList<Lechuga> lechugas;
15
16     public RegistroLechugas() {
17
18         lechugas = new ArrayList<>();
19     }
20
21     // Método para agregar una lechuga al registro
22     public void agregarLechuga(Lechuga lechuga) {
23
24         lechugas.add(lechuga);
25     }
26
27     // Método para guardar el registro en un archivo
28     public void guardarRegistro() {
29
30         try (PrintWriter writer = new PrintWriter(new FileWriter("registro_lechugas.txt"))) {
31
32             for (Lechuga lechuga : lechugas) {
33
34                 writer.println(lechuga.getInfo());
35             }
36         } catch (IOException e) {
37
38             e.printStackTrace();
39         }
40     }
41
42 }
43
44 }
45
```

Clase "Main":

```
1 private void verRegistroActionPerformed(java.awt.event.ActionEvent evt) {GEN-FIRST:event_verRegistroActionPerformed
2
3     // Especifica la ruta del archivo a abrir
4     String filePath = "registro_lechugas.txt"; // Ajusta el nombre del archivo según sea necesario
5
6     try {
7         // Crea un objeto File con la ruta del archivo
8         File file = new File(filePath);
9
10        // Verifica si el archivo existe antes de intentar abrirlo
11        if (file.exists()) {
12            // Abre el archivo con el programa predeterminado del sistema
13            Desktop.getDesktop().open(file);
14        } else {
15            JOptionPane.showMessageDialog(this, "El archivo no existe: " + filePath);
16        }
17    } catch (Exception ex) {
18        // Manejar cualquier excepción que pueda ocurrir al intentar abrir el archivo
19        ex.printStackTrace();
20    }
21 }
22 }
23 }
24 }
25
26 private void eliminarRegistroActionPerformed(java.awt.event.ActionEvent evt) {GEN-FIRST:event_eliminarRegistroActionPerformed
27
28     // Especifica la ruta del archivo a eliminar
29     String filePath = "registro_lechugas.txt"; // Ajusta el nombre del archivo según sea necesario
30
31     try {
32         // Crea un objeto File con la ruta del archivo
33         File file = new File(filePath);
34
35         // Verifica si el archivo existe antes de intentar eliminarlo
36         if (file.exists()) {
37             // Elimina el archivo
38             if (file.delete()) {
39                 JOptionPane.showMessageDialog(this, "El archivo se ha eliminado correctamente: " + filePath);
40             } else {
41                 JOptionPane.showMessageDialog(this, "El archivo no se pudo eliminar: " + filePath);
42             }
43         } else {
44             System.out.println("El archivo no existe: " + filePath);
45         }
46     } catch (Exception ex) {
47         // Manejar cualquier excepción que pueda ocurrir al intentar eliminar el archivo
48         ex.printStackTrace();
49     }
50 }
51 }
52 }
53 }
54 }
55
56 private void txtDayActionPerformed(java.awt.event.ActionEvent evt) {GEN-FIRST:event_txtDayActionPerformed
57 }
58 }
59 }
60
61 private void btnRegistrarActionPerformed(java.awt.event.ActionEvent evt) {GEN-FIRST:event_btnRegistrarActionPerformed
62
63     if (txtTemperature.getText().isEmpty() ||
64         txtHumedad.getText().isEmpty() ||
65         txtLongitud.getText().isEmpty() ||
66         txtDay.getText().isEmpty()) {
67         // Mostrar un JOptionPane indicando que se deben completar todos los campos
68         JOptionPane.showMessageDialog(this, "Por favor, complete todos los campos.", "Campos incompletos", JOptionPane.WARNING_MESSAGE);
69     } else {
70         // Si todos los campos están completos, proceder con el registro
71         // datos ingresados por el usuario
72         int id = jComboBox.getSelectedIndex() + 1; // Obtener el índice seleccionado y sumar 1 para obtener el ID
73         String fechaPlantacion = txtDay.getText();
74         double temperatura = Double.parseDouble(txtTemperature.getText());
75         double humedad = Double.parseDouble(txtHumedad.getText());
76         String longitud = txtLongitud.getText();
77
78         // Crear una instancia de Lechuga con los datos ingresados
79         Lechuga lechuga = new Lechuga(id, fechaPlantacion, temperatura, humedad, longitud);
80
81         // Agregar la lechuga al registro
82         registro.agregarLechuga(lechuga);
83
84         // Guardar el registro en un archivo
85         registro.guardarRegistro();
86
87         // Vaciar campos después de guardar
88         txtDay.setText("");
89         txtTemperature.setText("");
90         txtHumedad.setText("");
91         txtLongitud.setText("");
92     }
93 }
94 }
95
96 private void vaciarCamposActionPerformed(java.awt.event.ActionEvent evt) {GEN-FIRST:event_vaciarCamposActionPerformed
97
98     txtDay.setText("");
99     txtTemperature.setText("");
100    txtHumedad.setText("");
101    txtLongitud.setText("");
102 }
103 }
104
105 /**
106  * @param args the command line arguments
107  */
108 public static void main(String args[]) {
109
110     try {
111         for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
112             if ("Nimbus".equals(info.getName())) {
113                 javax.swing.UIManager.setLookAndFeel(info.getClassName());
114                 break;
115             }
116         }
117     } catch (ClassNotFoundException ex) {
118         java.util.logging.Logger.getLogger(Formulario.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
119     } catch (InstantiationException ex) {
120         java.util.logging.Logger.getLogger(Formulario.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
121     } catch (IllegalAccessException ex) {
122         java.util.logging.Logger.getLogger(Formulario.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
123     } catch (javax.swing.UnsupportedLookAndFeelException ex) {
124         java.util.logging.Logger.getLogger(Formulario.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
125     }
126 }
127
128 /**
129  * Create and display the form
130  */
131 public static void main(String args[]) {
132     java.awt.EventQueue.invokeLater(new Runnable() {
133         public void run() {
134             new Formulario().setVisible(true);
135         }
136     });
137 }
138 }
```

Interfaz de usuario

Control de siembras de lechugas

Opciones

¡Hola, bienvenido!

Id. de la lechuga: Lechuga normal ▼

Introduce el día:

Introduce la temperatura:

Introduce la humedad:

Introduce la longitud:

Registrar

Control de siembras de lechugas

Opciones

- Ver registros Ctrl+R
- Vaciar campos Ctrl+T
- Eliminar todos los registros Ctrl+Alt+E

¡Hola, bienvenido!

¡Hola, bienvenido!

Campos incompletos

Por favor, complete todos los campos.

OK

Referencias

De Roer, D. D., & De Roer, D. D. (2022, June 12). Clases FileReader y FileWriter para ficheros de texto en Java. *Disco Duro de Roer* -.

<https://www.discoduroderoer.es/clases-filereader-y-filewriter-para-ficheros-de-texto-en-java/>

Cálculo de volumen de la lechuga

Para diseñar una función generatriz que modele una lechuga de forma roseta con sólidos de revolución, podemos utilizar una función que describa la forma y las dimensiones de la lechuga en función de un parámetro. Aquí tienes un ejemplo:

Función generatriz de la lechuga:

$$f(t) = r(t) * h(t)$$

Donde:

$f(t)$ es la función generatriz que representa la lechuga.

$r(t)$ es una función que describe el radio de la lechuga en función del parámetro t .

$h(t)$ es una función que describe la altura de la lechuga en función del parámetro t .

Dentro de la función generatriz, las funciones $r(t)$ y $h(t)$ pueden ser definidas de diferentes maneras según las características deseadas para la lechuga. Aquí hay un ejemplo:

Lechuga de forma roseta:

$$r(t) = R_0 * \sin(t)$$

$$h(t) = H * \cos(t)$$

Donde:

R_0 es el radio máximo de la lechuga en la base.

H es la altura total de la lechuga.

En este ejemplo, utilizamos funciones trigonométricas para modelar la forma de la lechuga de forma roseta. La función $r(t)$ utiliza la función seno para crear la forma de los pétalos de la roseta, mientras que la función $h(t)$ utiliza la función coseno para darle una altura variante a la lechuga.

La función generatriz representa la geometría del sólido de revolución generado al rotar la curva definida por $f(t)$ alrededor de un eje.

Si utilizamos librerías de métodos numéricos en Java, podremos obtener el siguiente código para crear la grafica y revisar los resultados:

```
1 import org.jfree.chart.ChartFactory;
2 import org.jfree.chart.ChartFrame;
3 import org.jfree.chart.plot.PlotOrientation;
4 import org.jfree.data.xy.DefaultXYDataset;
5
6 import java.util.function.Function;
7
8 public class VolumenLechuga {
9     public static void main(String[] args) throws Exception {
10
11         //Definir la funcion para el radio de la lechuga
12         Function<Double, Double> radio = t -> 3 + 2 * Math.sin(2, t);
13
14         // Definir los limites de la integracion
15         double limiteInferior = 0.0;
16         double limiteSuperior = Math.PI / 4.0;
17
18         // Calcular el volumen exacto utilizando el metodo de discos
19         double volumenExacto = calcularVolumenExacto(radio, limiteInferior, limiteSuperior);
20
21         // Imprimir el resultado
22         System.out.println("El volumen exacto de la lechuga es: " + volumenExacto + "cm³");
23
24         // Crear los datos para el grafico
25         DefaultXYDataset dataset = crearDataset(radio, limiteInferior, limiteSuperior);
26
27         // Crear el grafico
28         ChartFrame frame = crearGrafico(dataset);
29
30         // Mostrar el grafico
31         frame.pack();
32         frame.setVisible(true);
33     }
34
35     public static double calcularVolumenExacto(Function<Double, Double> radio, double limiteInferior, double limiteSuperior) {
36
37         // Definir la precision para el calculo numerico
38         int n = 1000;
39
40         // Calcular el incremento en altura
41         double deltaT = (limiteSuperior - limiteInferior) / n;
42
43         // Calcular el volumen aproximado
44         double volumen = 0.0;
45         for(int i = 0; i < n; i++){
46             double t = limiteInferior + i * deltaT;
47             double r = radio.apply(t);
48             double area = Math.PI * Math.pow(r, 2);
49             volumen += area * deltaT;
50         }
51
52         return volumen;
53     }
54
55     public static DefaultXYDataset crearDataset(Function<Double, Double> radio, double limiteInferior, double limiteSuperior){
56
57         DefaultXYDataset dataset = new DefaultXYDataset();
58
59         // Numero de puntos en el grafico
60         int n = 100;
61         double [][] data = new double[2][n];
62         double deltaT = (limiteSuperior - limiteInferior) / (n-1);
63
64         for(int i = 0; i < n; i++){
65             double t = limiteInferior + i * deltaT;
66             double r = radio.apply(t);
67             data[0][i] = t;
68             data[1][i] = r;
69         }
70
71         dataset.addSeries("Radio", data);
72
73         return dataset;
74     }
75
76     public static ChartFrame crearGrafico(DefaultXYDataset dataset){
77
78         String titulo = "Grafico de la lechuga";
79         String etiquetaX = "Altura(t)";
80         String etiquetaY = "Radio(r)";
81
82         // Crear el grafico de dispersion
83         org.jfree.chart.JFreeChart chart = ChartFactory.createScatterPlot(titulo, etiquetaX, etiquetaY, dataset,
84             PlotOrientation.VERTICAL, true, true, false);
85
86         ChartFrame frame = new ChartFrame("Grafico de la lechuga", chart);
87
88         return frame;
89     }
90 }
```


Imagen del programa (Graficando):

Conclusiones